**Elementos Arquitectónicos de Software para la Integración de Frecuencias de Enfermedades**

*Software Architectural Elements for the Integration of Diseases Frequency*

Paul Mendoza del Carpio \*

**RESUMEN**

La detección temprana de amenazas de propagación de enfermedades infecciosas ha sido un tema de interés, por su papel importante en la mejora de la efectividad de respuesta. Ella proporciona un repositorio de datos fértil para el alerta, y una guía de planificación, implementación y evaluación de programas para la prevención y control de enfermedades.

Los registros de datos de salud de pacientes de un departamento o institución pueden ser resumidos para estudios epidemiológicos de larga escala. Luego, la integración de registros desde múltiples centros clínicos con el propósito de obtención de resultados relevantes es una característica necesaria en sistemas de salud regional o nacional.

El presente trabajo presenta un conjunto de elementos arquitectónicos basados en tecnología middleware para la integración de frecuencias de enfermedades identificadas según la clasificación internacional CIE-10 tomando en consideración características de localización y agregación de datos.

**Palabras clave:** *arquitectura de software, CIE-10, middleware.*

**ABSTRACT**

The early detection of threats in infectious diseases propagation has been a topic of interest due to its important role in the improvement of an effective response. This brings a rich data repository for alerts, and a guide of planning, implementation and assessment of programs for the prevention and control of diseases.

The health records of patients of regions or institutions can be resumed for epidemiologic analysis of large scale. Then, the integration of records from multiple clinical centers with the purpose of getting relevant results is a required feature in regional or national health systems.

This work presents a set of architectural elements based on middleware technology for the integration of diseases frequency identified following the international classification ICD-10 considering features of localization and data aggregation.

**Key words:** *software architecture, ICD-10, middleware.*

\* Magister en Ingeniería de Software y Candidato a Doctor en Ciencias de la Computación. Profesional IBM Certificado. Docente de la UAP.

 E-mail: pmendozadelcarpio@gmail.com

 Twitter: @paulnetedu

**INTRODUCCIÓN**

La detección temprana de amenazas de propagación de enfermedades infecciosas y bioterrorismo ha sido un tema de interés durante los últimos años [7] [12] [3] [11] [3] [18]. El temprano conocimiento de la propagación de enfermedades juega un papel importante en la mejora de la efectividad de respuesta, el cual se puede ver retrasado entre otros motivos a una labor intensiva de transcripción para la recopilación y centralización de datos [8] [14]. La recopilación de la ocurrencia de enfermedades proporciona un repositorio de datos fértil, una comunicación activa para el alerta frente a situaciones, y una guía de planificación, implementación y evaluación de programas de largo término para la prevención y control de enfermedades (distribución de medicación, planes de vacunación, adjudicación de recursos) [7].

Las alianzas para la formación de redes de salud han presentado un creciente desarrollo los últimos años pero a su vez también se presenta una necesidad por una perspectiva más holística de las redes de salud [17]. Los registros de datos de salud de los pacientes de un departamento o institución pueden ser resumidos para estudios epidemiológicos de larga escala [8]. El bajo predominio de las condiciones de salud adecuadas reclama por la integración de registros desde múltiples centros clínicos con el propósito de obtención de resultados relevantes estadísticamente [8]. Se pueden encontrar varios sistemas de supervisión de enfermedades, los cuales comparten objetivos similares pero varían en su arquitectura y en su procesamiento y técnicas de administración de la información, ellos presentan diferente cobertura geográfica y de enfermedades (e.g. influenza, fiebre aftosa). A pesar que éstos sistemas han sido desplegados en ciudades, y estados, es necesario crear una infraestructura inter-jurisdiccional, inter-regional para maximizar el beneficio potencial de una supervisión de enfermedades global [7] [8] [14]. Los participantes llamados a establecer un mecanismo de vigilancia de enfermedades en una región son las instituciones de salud (originadores de datos) y autoridades de salud pública (receptores de datos y reactores frente a los datos recibidos) [12].

En el campo de salud se vienen desarrollando iniciativas para el uso de tecnologías de información y comunicación (TIC) en la actividad sanitaria. Iniciativas que surgen como solución ante algunos inconvenientes encontrados en las actividades no informatizadas. Por ejemplo, en el caso de las historias clínicas de papel se presenta el desorden, falta de uniformidad, ilegibilidad, alteración de la información, no disponibilidad [2], inconvenientes que pueden ser resueltos mediante un correcto empleo de las historias clínicas electrónicas (HCE). Tal informatización es de relevancia debido a que permite la integración de la información clínico-asistencial del paciente, así como el acceso a la misma desde lugares distantes geográficamente. Además, se debe señalar la alta importancia que merece la integración de datos en el campo de salud para en consecuencia formar un sistema integrado de información clínica.

**Hipótesis**

El uso de tecnología MOM (*Message Oriented Middleware*) es de relevancia para el desarrollo de elementos arquitectónicos de integración de datos.

**MATERIALES Y MÉTODOS**

**Material**

1. *Data de mortalidad:* data de mortalidad publicada por la OMS (Organización Mundial de la Salud) la cual es reportada anualmente por los países miembros en todo el mundo, tal data considera edad, sexo y causa de muerte.
2. *Software:* Servidor MOM IBM WebSphere MQ, servidor ESB Apache ServiceMix servidor de aplicaciones Oracle Sun Glassfish.
3. *Estándares:* CIE (Clasificación Internacional de Enfermedades), JMS (*Java Messaging Service*), XSD (XML Schema Definition), UML (*Unified Modeling Language*).

**METODOLOGÍA**

1. *Recolección de datos de mortalidad*

Se pobló una base de datos con datos de mortalidad publicados por la OMS. Los datos serán transmitidos a través de diferentes servidores de software en diferentes tiempos, considerando los servidores iniciados y parados, esto último para mostrar la confiabilidad de recepción de mensajes.

1. *Envío asíncrono de mensajes*

Los mensajes han sido transmitidos de un servidor a otro en forma asíncrona, aquí se sacará provecho del uso de colas de mensajes (MQ).

1. *Concurrencia*

Se implementa hilos de ejecución para emular procesos que publican en forma simultánea los datos de frecuencias de enfermedades desde diferentes orígenes de datos.

**RESULTADOS**

Los resultados producto del presente trabajo son los siguientes:

1. *Esquema de datos*

Con el fin de alcanzar un buen matching de la información proporcionada por cada institución de salud se desarrolló un formato de datos único a emplear por cada origen de datos en el envío de su información, este consideró como base información de reportes anuales de aseguradoras de salud y el esquema de datos empleado por el API de geocodificación de Google. El formato propuesto consiste en un esquema XML que es presentado en forma gráfica en los siguientes párrafos.

La raíz del esquema XML se denomina *fint-disease*, y está conformada por una secuencia de casos (*case*) como se muestra en la figura 1. Cada caso cuenta con un identificador (*id*), un identificador de la petición de los datos (*requestId*), un proveedor de la información (*provider*), datos propiamente de la enfermedad (*disease*), datos relacionados al tiempo en que se presentó el caso (*time*), datos de la población involucrada (*population*), localización (*place*), categorización que se desee dar para el caso (*categoryId*), y un elemento disponible para otros datos datos que se deseen agregar (*ext*).



**Figura 1.** Representación de casos de ocurrencias de enfermedades.

El elemento *provider* contiene datos del proveedor de la información: nombre (*name*) e identificador de red de salud (*networkId*). Véase la figura 2.



**Figura 2.** Representación de proveedores de la información.

El elemento *disease* contiene información propiamente de la enfermedad a la que hace referencia el caso: identificador (id, código de la enfermedad), sistema de codificación (*codeSystem*, e.g. CIE-10), identificador del tipo (*typeId*, e.g. presuntivo, definitivo), método de detección (*detectionMethodId*), modo de transmisión (*transmissionModeId*), código importado (*importedCode*, código importado de otro sistema). Véase la figura 3.



**Figura 3.** Representación de información de la enfermedad.

Acerca de información del intervalo de tiempo en que se dio el caso de la enfermedad, se considera el tiempo en que la información es registrada (*availability*) y el tiempo efectivo en que se habría presentado efectivamente la enfermedad (*effective*) como se muestra en la figura 4.



**Figura 4.** Representación de información temporal.

El elemento *population* está relacionado a información acerca de la población afecta a la enfermedad: identificador del servicio (*careServiceId*), número de personas afectadas (*nomberOfPeople*), incidencia (*incidenceSize*), prevalencia (*prevalenceSize*), mortalidad (*mortalitySize*), sexo (*sex*) y edad (*age*). Véase la figura 5.



**Figura 5.** Representación de información de la población.

El elemento place contempla información acerca de la localización en que se presentó el caso, aquí se tomó como referencia el esquema de datos empleado la API de geocodificación de Google: identificador del país (*countryId*), identificador del área administrativa (*administrativeAreaId*), locación (*location*, con latitud y longitud), dirección (*address*). Véase la figura 6.



**Figura 6.** Representación de la ubicación.

*B. Componentes*

A continuación se da un resumen de los principales componentes relevantes para la arquitectura de integración (véase también la figura 7):

1. Marshaller: componente para la transformación de objetos a XML (*marshalling*) y viceversa *unmarshalling*). El archivo XML en cuestión sigue el esquema presentado en la sección anterior.
2. App Enviador Mensajes: aplicación que envía mensajes como un cliente JMS. Éstos mensajes representan cada caso de enfermedad presentado en una institución de salud.
3. Cola de mensajes: destino de mensajes XML que representan los casos proporcionados por cada origen de dato mediante a un App Enviador Mensajes.
4. Ruteador de mensajes: componente que mantiene las reglas de ruteo de mensajes, encargado de aplicar agregación sobre los mensajes proporcionados por cada institución de salud.
5. Cola de mensajes agregados: destino de mensajes XML resumidos, producto de haber aplicado estrategias de agregación en el ruteador de mensajes.
6. App Receptor Mensajes: aplicación que recibe mensajes agregados de una cola de mensajes. Puede ser implementado como un cliente JMS o un MDB (*Message Driven Bean*).



**Figura 7.** Componentes de la arquitectura.

*C. Proceso*

Se presenta un proceso simplificado de integración de casos de enfermedades, el cual sigue la siguiente secuencia según los entornos de ejecución mostrados en la figura 8:

1. Cada institución de salud envía los casos de enfermedades presentados en su establecimiento de salud hacia una entidad autoridad de salud.
2. La autoridad de salud recibe los casos presentados por las instituciones de salud.
3. El entorno de la autoridad de salud aplica agregación de datos para luego obtener casos de resumen.
4. La autoridad de salud envía los casos agregados al organismo de salud.
5. El organismo de salud recibe los casos de enfermedades agregados.



**Figura 8.** Entornos de ejecución.

Según se muestra en la figura 8, la arquitectura contempla el uso de un ESB (*Enterprise Service Bus*) el cuál atenderá una añadidura no dificultosa de nuevos sistemas mediante el uso de reglas de ruteo configurables en el servidor. La figura 9 muestra las reglas de ruteo actualmente consideradas en la arquitectura, nótese la ejecución de agregación (*aggregate*) y el envío de mensajes agregados a múltiples destinos posibles (*multicast*).



**Figura 9.** Ruteo de mensajes.

Las reglas de ruteo de mensajes están implementadas en Apache Camel, en la figura 10 se muestra el descriptor XML correspondiente a la representación gráfica de la figura 9.



**Figura 10.** Documento XML de ruteo en Apache Camel.

*D. Patrones de software*

Los aplicativos de las entidades de salud pueden publicar sus casos de enfermedades al sistema de integración mediante un adaptador (patrón *Channel Adapter*) que permitirá el envío de mensajes. Para el envío se implementará un temporizador que ejecute periódicamente el propio envío, y una clase que convierta la representación de caso de enfermedad (que utiliza actualmente la entidad) en un mapa clave-valor. Esta última clase encargada de convertir la representación será empleada por el framework para elaborar el mensaje en el formato actual del sistema de integración.

Considerando que los formatos de mensajes pueden evolucionar en nuevas versiones, se propone emplear un canal distintivo para cada tipo de formato, de esta forma cada receptor del mensaje podrá conocer en forma específica cómo dar tratamiento al mensaje (patrón *Datatype Channel*). Así mismo, con el transcurrir del tiempo algunos formatos de amplio uso pueden más adelante sólo ser empleados por algunas entidades de salud, frente a ello éstas entidades pueden continuar publicando sus casos en el mismo destino, sin embargo el sistema de integración empleará un traductor interno (*Message translator*) para modificar el formato hacia el formato vigente.



**Figura 11.** Canales de mensajes.

Para orígenes de datos NoSQL se aplicará el patrón *MapReduce Partitioning* para mover los registros de enfermedades en categorías, de esta forma más tarde se tendrá un menos trabajoso envío de mensajes de casos. Algunas bases de datos NoSQL trabajan con el formato JSON para la representación de datos, para ellas u orígenes de datos no estructurados se implementaría un *Adapter* para enviar sus mensajes en el formato XML estándar empleado al momento, el framework cuenta con *Adapters* ya implementados para algunas bases de datos NoSQL (MongoDB, Neo4j).

El asumir que las diversas entidades de salud participantes empleen el mismo sistema de codificación de enfermedades puede resultar poco práctico y en algunos casos resulta irreal (CIE-10, CIE-9, SNOMED). El sistema de codificación empleado por cada entidad debe estar establecido en el contenido de sus mensajes, para ellos se hará uso de un ruteador (patrón *Content-based router*), el cual basado en el sistema de codificación empleado en el mensaje, direccionará éste al canal adecuado. Así mismo se emplea el patrón *Routing Slip* para direccionar los mensajes en forma dinámica.



**Figura 12.** Ruteo de mensajes según codificación.

Una vez que los mensajes se encuentran en una sola codificación se procede a aplicar un algoritmo de agregación de data (patrón *Aggregator*), de esta forma la información es resumida para su uso por algún organismo de salud, véase la figura 8. Luego se emplea un canal que entrega una copia del mensaje agregado a cada uno de sus suscriptores (patrón *Publish-Subscribe Channel*), con ello cada mensaje agregado podría ser empleado para diversos propósitos: ser almacenados, ser consumidos por otros organismos o entidades, otro (de acuerdo al suscriptor).



**Figura 13.** Agregación de mensajes.

Mensajes no válidos pueden presentarse en el proceso de integración debido a diferentes motivos: documentos XML mal formados, codificación de enfermedades incorrectamente establecidas, o la creación errónea de mensajes agregados (al aplicar un nuevo algoritmo de agregación). Para poder manejar estos casos se hace uso de canales destino de mensajes inválidos (patrón *Invalid Message Channel*).

**ANÁLISIS Y DISCUSIÓN**

El software catalogado como middleware sirve como puente entre las diferentes aplicaciones involucradas en la integración de sistemas. Emplear una capa de interfaz común en lugar de una integración punto-a-punto permite a todas las aplicaciones involucradas el comunicarse libremente con cualquier otra. Luego, es posible la reutilización de aplicaciones existentes y la incorporación de nuevas aplicaciones y data relacionada.

El bus de servicios empresarial o comúnmente denominado ESB es una solución que permite un confiable acceso y traslado de información dentro de una empresa y fuera de ella en forma rápida y flexible haciendo uso de un motor de entrega y transformación de datos. Es así que las capacidades del ESB pueden ser empleadas en empresas de atención médica para la comunicación, mediación y transformación de datos desde cualquier origen hacia diferentes plataformas de destino, cuando sea necesario y en el formato de datos que se requiera.



**Figura 14.** Enterprise Service Bus.

Se tienen características técnicas importantes que un ESB puede aportar para la integración empresarial en atención de salud:

* Lectura, escritura, actualización sobre bases de datos Oracle, PostgreSQL, DB2, SQL Server, entre otras ampliamente utilizadas.
* Soporte de formatos de datos XML como HL7 V3, CDA (*Clinical Document Architecture*), CCR (*Continuity of Care Record*).
* Consumo de Web Services.
* Ejecución de operaciones PUT y GET sobre colas de mensajería y compatibilidad con JMS.
* Soporte de protocolo HTTP.
* Soporte de protocolo SOAP.
* Soporte para aplicaciones SAP.

Los componentes en el presente trabajo fueron implementados y probados para la integración de casos de enfermedades asociados a data de mortalidad de diversos países, data obtenida y disponible por la OMS. La ejecución mostró efectivamente que mediante el uso de MOM es posible la transmisión de mensajes de datos entre sistemas aun cuando los sistemas no se encuentran iniciados (activos), realizándose la recepción de los mensajes. Así mismo el lanzamiento de hilos de ejecución presentó un correcto procesamiento de los mensajes en un escenario concurrente. Cada nodo de ejecución mostró capacidad de procesamiento al recibir múltiples peticiones en forma simultánea.

Se debe tener en consideración que cada proveedor de ESB presenta diferentes componentes y en algunos casos diferentes lenguajes para el procesamiento de mensajes y data, por ello es necesario conocer y evaluar la funcionalidad y rendimiento del componente candidato a emplear en una solución de integración. Se recomendaría en un trabajo futuro realizar una comparativa de rendimiento entre diferentes productos MOM y ESB presentes en el mercado.

**CONCLUSIONES**

1. Los elementos arquitectónicos presentados atienden características necesarias de una solución de integración como son el buen matching de información, el bajo acoplamiento entre orígenes de datos, y la añadidura no dificultosa.
2. La presencia de tecnología MOM (Message Oriented Middleware) y ESB son de importancia para garantizar la comunicación de mensajes y permitir añadir nuevos elementos en las reglas de ruteo de mensajes respectivamente.

**REFERENCIAS BIBLIOGRÁFICAS**

[1] Aschengrau, A., and Seage, G. Essentials of epidemiology in public health. Epidemiology Series. Jones and Bartlett, 2003.

[2] Azcárate, J. “De la historia clínica a la historia de salud electrónica: Pamplona, 18 de diciembre de 2003”. Informes SEIS. Sociedad Española de Informática de la Salud, 2003.

[3] Bauer, Jr., D. W., and Mohtashemi, M. An application of parallel monte carlo modeling for real-time disease surveillance. In Proceedings of the 40th Conference on Winter Simulation (2008), WSC 08, Winter Simulation Conference, pp. 1029{1037.

[4] Bhopal, R. Concepts of epidemiology: integrating the ideas, theories, principles, and methods of epidemiology. Oxford University Press, 2008.

[5] Bonita, R., Beaglehole, and Kjellstrom, T. “Basic epidemiology”, 2nd ed. ed. World Health Organization, Geneva, 2006.

[6] Carr, S., Unwin, N., and Pless-Mulloli, T. An Introduction to Public Health and Epidemiology, second edition ed. Open University Press, 2007.

[7] Chen, H., Zeng, D., Buckeridge, D. L., Izadi, M. I., Verma, A., Okhmatovskaia, A., Hu, X., Shen, X., Cao, Z., Wang, F.-Y., Zheng, X., and Wang, Q. “AI for global disease surveillance”. IEEE Intelligent Systems 24 (November 2009), 66-82.

[8] Elger, B. S., Iavindrasana, J., Lo Iacono, L., Muller, H., Roduit, N., Summers, P., and Wright, J. Strategies for health data exchange for secondary, cross-institutional clinical research. Comput. Methods Prog. Biomed. 99 (September 2010).

[9] Fewtrell, L., Bartram, J., and Organization, W. H. Water quality: guidelines, standards, and health : assessment of risk and risk management for water-related infectious disease. World Health Organization Water. IWA Pub., 2001.

[10] Hohpe, G., and Woolf, B. “Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions”. Boston, MA, USA : Addison-Wesley Longman Publishing Co., Inc., 2003. – ISBN 0321200683

[11] Lopes, L. F., Silva, F. A. B., Couto, F., Zamite, J. a., Ferreira, H., Sousa, C., and Silva, M. J. Epidemic marketplace: an information management system for epidemiological data. In Proceedings of the First international conference on Information technology in bio- and medical informatics (Berlin,

Heidelberg, 2010), ITBAM'10, Springer-Verlag, pp. 31.

[12] Mandl, K. D., Overhage, J. M., Wagner, M. M., Lober, W. B., Sebas tiani, P., Mostashari, F., Pavlin, J. A., Gesteland, P. H., Treadwell, T., Koski, E., and et al. Implementing syndromic surveillance: A practical guide informed by the early experience. Journal of the American Medical Informatics Association 11, 2 (2004), 141.

[13] Mayfong, M. Measurement in epidemiology: Frequency, association, and impact. Training Course in Reproductive Health Research, October 2009.

[14] Odero, W., Rotich, J., Yiannoutsos, C. T., Ouna, T., and Tierney, W. M. Innovative approaches to application of information technology in disease surveillance and prevention in western Kenya. J. of Biomedical Informatics 40 (August 2007).

[15] Rockett, I. Population and health: An introduction to epidemiology. Population Bulletin 54, 4 (December 1999).

[16]Silva, I. Cancer epidemiology: principles and methods. Iarc Nonserial Publication. International Agency for Research on Cancer, 1999.

[17] Thrasher, E. H., Craighead, C. W., and Byrd, T. A. An empirical investigation of integration in healthcare alliance networks. Decis. Support Syst. 50 (December 2010), 116.

[18] Zamite, J. a., Silva, F. A. B., Couto, F., and Silva, M. J. Medcollector: multisource epidemic data collector. In Proceedings of the First international conference on Information technology in bio- and medical informatics (Berlin, Heidelberg, 2010), ITBAM'10, Springer-Verlag, pp. 16.